

SSSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSSS	LLL	0000000000	AAAAAAA
SSSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSSS	LLL	0000000000	AAAAAAA
SSSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSSS	LLL	0000000000	AAAAAAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSSSSSSSSS	YYY	YYY	SSSSSSSSSS	LLL	000	000 AAA AAA
SSSSSSSSSS	YYY	YYY	SSSSSSSSSS	LLL	000	000 AAA AAA
SSSSSSSSSS	YYY	YYY	SSSSSSSSSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAAA AAAAA
SSS	YYY	YYY	SSS	LLL	000	000 AAAA AAAAA
SSS	YYY	YYY	SSS	LLL	000	000 AAAA AAAAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSS	YYY	YYY	SSS	LLL	000	000 AAA AAA
SSSSSSSSSS	YYY	SSSSSSSSSS	LLLLLLLLLLLL	0000000000	AAA	AAA
SSSSSSSSSS	YYY	SSSSSSSSSS	LLLLLLLLLLLL	0000000000	AAA	AAA
SSSSSSSSSS	YYY	SSSSSSSSSS	LLLLLLLLLLLL	0000000000	AAA	AAA

Sy  
--  
SS1  
SS1  
SS1  
SS1  
SS1  
SS1  
SYS  
SYS  
SYS  
TRY  
UNL  
WR]

000000	PPPPPPPP	DDDDDDDD	RRRRRRRR	VV	VV	77777777	9999999	000000
000000	PPPPPPPP	DDDDDDDD	RRRRRRRR	VV	VV	77777777	9999999	000000
00	00 PP	PP DD	DD RR	RR VV	VV	77	99	00
00	00 PP	PP DD	DD RR	RR VV	VV	77	99	00
00	00 PP	PP DD	DD RR	RR VV	VV	77	99	00
00	00 PP	PP DD	DD RR	RR VV	VV	77	99	00
00	00 PPPPPPPP	DD	DD RRRRRRRR	VV	VV	77	99999999	00
00	00 PPPPPPPP	DD	DD RRRRRRRR	VV	VV	77	99999999	00
00	00 PP	DD	DD RR RR	VV	VV	77	99	00
00	00 PP	DD	DD RR RR	VV	VV	77	99	00
00	00 PP	DD	DD RR RR	VV	VV	77	99	00
00	00 PP	DD	DD RR RR	VV	VV	77	99	00
000000	PP	DDDDDDDD	RR	RR	VV	999999	000000	....
000000	PP	DDDDDDDD	RR	RR	VV	999999	000000	....

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSSS
LL		SSSSSS
LL		SS
LL		SS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

(3)	210	CONSOLE RECEIVE DISPATCH VECTOR
(4)	229	CONSOLE CONTROLLER INITIALIZATION
(5)	276	CONSOLE UNIT INITIALIZATION
(6)	344	CON\$DISCONNECT DISCONNECT LINE
(7)	369	OUTPUT MODEM CONTROL
(8)	403	CONSOLE RECEIVER INTERRUPT DISPATCHER
(9)	482	ENVIRONMENTAL MONITOR INPUT INTERRUPT
(10)	615	LOGICAL CONSOLE INPUT INTERRUPTS
(11)	645	START I/O ON CONSOLE INTERFACE
(12)	693	CARRIER CHANGE SENT BY CONSOLE
(13)	755	CONSOLE TRANSMITTER INTERRUPT SERVICE
(14)	891	CONSOLE PORT ACTION ROUTINES
(15)	950	CON\$SENDCONSCMD - Send command to 11/790 console
(16)	1024	"ALLOCATE" CONSOLE TERMINAL
(17)	1075	RELEASE CONSOLE TERMINAL
(18)	1104	CON\$GETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL
(19)	1129	CON\$PUTCHAR - PUT A CHARACTER TO THE CONSOLE TERMINAL

0000 1 .TITLE OPDRV790 - VAX/VMS 11/790 CONSOLE TERMINAL DRIVER  
0000 2 .IDENT 'V04-000'  
0000 3 \*\*\*\*\*  
0000 4 \*\*\*\*\*  
0000 5 \*  
0000 6 \* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0000 7 \* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0000 8 \* ALL RIGHTS RESERVED.  
0000 9 \*  
0000 10 \* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0000 11 \* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0000 12 \* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0000 13 \* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0000 14 \* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0000 15 \* TRANSFERRED.  
0000 16 \*  
0000 17 \* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0000 18 \* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0000 19 \* CORPORATION.  
0000 20 \*  
0000 21 \* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0000 22 \* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0000 23 \*  
0000 24 \*  
0000 25 \*\*\*\*\*  
0000 26  
0000 27 ++  
0000 28 FACILITY:  
0000 29  
0000 30 VAX/VMS I/O SUBSYSTEM  
0000 31  
0000 32 ABSTRACT:  
0000 33  
0000 34  
0000 35 AUTHOR: Benn Schreiber, Trudy Matthews  
0000 36  
0000 37 MODIFIED BY:  
0000 38  
0000 39 V03-011 TCM0009 Trudy C. Matthews 20-Aug-1984  
0000 40 Fix input interrupt dispatching through the dispatch table.  
0000 41  
0000 42 V03-010 TCM0008 Trudy C. Matthews 31-Jul-1984  
0000 43 Move routine CON\$KEEPALIVE to module MCHECK790.  
0000 44  
0000 45 V03-009 TCM0007 Trudy C. Matthews 24-Jul-1984  
0000 46 Fix some undefined symbols in TCM0006.  
0000 47  
0000 48 V03-008 TCM0006 Trudy C. Matthews 19-Jul-1984  
0000 49 Add routine CON\$KEEPALIVE, which is called periodically to  
0000 50 determine if the VENUS console software is still functioning.  
0000 51 Update the CONSC\_xxx console function code definitions.  
0000 52 Add TXDBSC\_xxx and RXDBSC\_xxx definitions for EMM and logical  
0000 53 console lines.  
0000 54  
0000 55 V03-007 WHM0001 Bill Matthews 09-Jul-1984  
0000 56 Add routines CON\$PUTCHAR, CON\$GETCHAR, and CON\$INIT\_CTY to do  
0000 57 non-interrupt driven I/O to the console terminal.

45  
65  
2D  
74  
68  
20  
6F  
6C  
74  
676D  
64  
20  
6E  
72

0000	58	
0000	59	V03-006 TCM0005 Trudy C. Matthews 20-Jun-1984
0000	60	Add a first pass at handling venus environmental monitor
0000	61	alerts.
0000	62	
0000	63	V03-005 TCM0004 Trudy C. Matthews 04-Jun-1984
0000	64	Track changes in the console interface protocol: we can not
0000	65	read or write TXCS unless the "READY" bit is set; also, the
0000	66	"write enable" bit position has changed. Add routine
0000	67	CONSRELEASECTY.
0000	68	
0000	69	V03-004 TCM0003 Trudy C. Matthews 09-Apr-1984
0000	70	Add dummy entry point for CONSRELEASECTY. The full routine
0000	71	will be added later.
0000	72	
0000	73	V03-003 TCM0002 Trudy C. Matthews 13-Dec-1983
0000	74	Add support for the "write enable bit" for line enable mask
0000	75	in VENUS' TXCS in routines CON\$SENDCONSCMD and CON\$OWNCTY.
0000	76	Modify interface to CON\$OWNCTY: it now returns the values
0000	77	that should be restored to TXCS and RXCS when the caller
0000	78	is done with "exclusive" use of the console terminal.
0000	79	
0000	80	V03-002 MIR0084 Michael I. Rosenblum 03-Aug-1983
0000	81	Change references to TT_CARRIER and TT_DTR to
0000	82	DZ. Remove STOP2 entry point and all references to
0000	83	STOP2.
0000	84	
0000	85	V03-001 TCM0001 Trudy C. Matthews 25-Mar-1983
0000	86	Modify CON\$OWNCTY to work with 11/790 console interface.
0000	87	
0000	88	-

```

0000 90 : SYMBOL DEFINITIONS
0000 91 :
0000 92 :   ; SYMBOL DEFINITIONS
0000 93 :
0000 94 : $CONDEF           ; CONSOLE DEFINITIONS
0000 95 : $CRBDEF          ; DEFINE CRB
0000 96 : $DCDEF            ; DEFINE DEVICE CLASSES
0000 97 : $DDBDEF          ; DEFINE DDB
0000 98 : $DEVDEF           ; DEFINE DEVICE CHARACTERISTICS
0000 99 : $DPTDEF           ; DRIVER PROLOGUE TABLE
0000 100: $DYNDEF          ; STRUCTURE TYPE CODE DEFINITIONS
0000 101: $EMBDEF <HD>     ; DEFINE ERRORLOG ENTRY FORMATS
0000 102: $IDBDEF          ; DEFINE IDB
0000 103: $IPLDEF           ; DEFINE IPL LEVELS
0000 104: $IRPDEF           ; DEFINE IRP OFFSETS
0000 105: $PRDEF             ; DEFINE PROCESSOR REGISTERS
0000 106: $PR790DEF         ; DEFINE 11/790 PROCESSOR REGISTERS
0000 107: $TTDEF             ; DEFINE TERMINAL CHARACTERISTICS
0000 108: $UCBDEF           ; DEFINE UCB
0000 109: $TTYDEFS          ; TTY UCB extension(must FOLLOW SUCBDEF)
0000 110: $TTYMODEM          ; TTY modem definitions
0000 111: $TTYMACS          ; TTY macro definitions
0000 112: $VECDEF            ; DEFINE CRB VECTOR
0000 113: $WCBDEF           ; Define WCB

0000 114 : TXCS REGISTER DEFINITIONS
0000 115 : $DEFINI TXCS
0000 116 :
0000 117 : $DEFINI TXCS
0000 118 :
0000 119 $DEF TXS .BLKL 1 ; TXCS REGISTER
0004 120 :
0004 121 : _VIELD TXS,0,<- ; DEFINE FIELDS TXS_M_XXX
0004 122 : <6>,-
0004 123 : <IE,,M>,- ; MBZ BITS
0004 124 : <RDY,,M>,- ; INTERRUPT ENABLE BIT
0004 125 : <ID,4,M>,- ; READY BIT
0004 126 : <,3>,-
0004 127 : <WRTENA,,M>,- ; ID FIELD
0004 128 : <TEM,8,M>,- ; MBZ BITS
0004 129 : >                ; ENABLE WRITE TO TXCS MASK FIELD
0004 130 : .=TXS            ; TRANSMITTER ENABLE MASK
0000 131 : _VIELD TXS,0,<- ; TRANSMITTER ENABLE MASK BITS
0000 132 : <16>,-
0000 133 : <CONSTTERM,,M>,- ; SKIP TO TRANSMIT ENABLE MASK FIELD
0000 134 : <REMPORT,,M>,- ; CONSOLE TERMINAL
0000 135 : <EMM,,M>,- ; REMOTE SERVICES PORT
0000 136 : <LOGCONS,,M>,- ; ENVIRONMENTAL MONITOR
0000 137 : >                ; LOGICAL CONSOLE
0000 138 : $DEFEND TXCS
0000 139 :
0000 140 : $DEFEND TXCS
0000 141 : : DEFINE CONSTANTS FOR TXDB DATA:
0000 142 : :
0000 143 : : THE FOLLOWING VALUES CAN BE SENT ON THE EMM LINE.
0000 144 : :
0000 145 : TXDBSC_EMM_STATUS = 0 ; REQUEST FOR EMM STATUS
0000 146 : TXDBSC_EMM_ENV = 1  ; REQUEST FOR SYSTEM ENVIRONMENT INFO

```

```

00000010 0000 147 TXDBSC_EMM_MARGIN = 16 :COMMAND TO SET MARGIN REGULATORS
00000011 0000 148 TXDBSC_EMM_CANCEL = 17 :CANCEL EMM REQUESTS ON INTERFACE
00000010 0000 149 :
00000011 0000 150 ; THE FOLLOWING VALUES CAN BE SENT ON THE LOGICAL CONSOLE LINE.
00000010 0000 151 :
00000011 0000 152 TXDBSC_LOG_EXWARM = ^X10 :EXAMINE WARMSTART FLAG
00000011 0000 153 TXDBSC_LOG_EXCOLD = ^X11 :EXAMINE COLDSTART FLAG
00000012 0000 154 TXDBSC_LOG_EXUCODE = ^X12 :EXAMINE MICROCODE VERSION LEVEL
00000020 0000 155 TXDBSC_LOG_SNDDIAG = ^X20 :SEND DIAGNOSTIC COMMAND STRING
00000030 0000 156 TXDBSC_LOG_REQERL = ^X30 :REQUEST ERRORLOG DATA
00000031 0000 157 TXDBSC_LOG_INVSNP1 = ^X31 :INVALIDATE ERRORLOG SNAPSHOT 1
00000032 0000 158 TXDBSC_LOG_INVSNP2 = ^X32 :INVALIDATE ERRORLOG SNAPSHOT 2
00000070 0000 159 TXDBSC_LOG_CANCEL = ^X70 :CANCEL CURRENT AND QUEUED REQUESTS
00000010 0000 160 :
00000011 0000 161 ; RXCS REGISTER DEFINITIONS
00000011 0000 162 :
00000011 0000 163 SDEFINI RXCS
00000011 0000 164 :
00000011 0000 165 SDEF RXS .BLKL 1 ;RXCS REGISTER
0004 166 _VIELD RXS,0,<->
0004 167 <6>,-
0004 168 <IE,M>,-
0004 169 <DONE,,M>,-
0004 170 <8>,-
0004 171 <DTR,8,M>,-
0004 172 >
0004 173 .=RXS
00000000 0004 174 _VIELD RXS,0,<->
00000000 0004 175 <16>,-
00000000 0004 176 <CONTERM,,M>,-
00000000 0004 177 <REMPORT,,M>,-
00000000 0004 178 <EMM,,M>,-
00000000 0004 179 <LOGCONS,,M>,-
00000000 0004 180 >
00000000 0004 181 SDEFEND RXCS
00000000 0004 182 :
00000000 0004 183 SDEFINI RXDB
00000000 0004 184 CONSOLE RXDB REGISTER
00000000 0004 185 :
00000000 0004 186 SDEFINI RXDB
00000000 0004 187 SDEF RXD .BLKL 1 ;RXDB RECEIVER DATA REGISTER
0004 188 _VIELD RXD,0,<->
0004 189 <DATA,8>,-
0004 190 <ID,4>,-
0004 191 <,4>,-
0004 192 <CARRIER,8>,-
0004 193 >
0004 194 SDEFEND RXDB
00000000 0004 195 :
00000000 0004 196 ; DEFINE DATA VALUES THAT CAN BE RECEIVED ON THE EMM AND LOGICAL CONSOLE
00000000 0004 197 ; LINES.
00000000 0004 198 :
00000000 0004 199 ; THE FOLLOWING RXDB VALUES MAY BE RECEIVED ON THE LOGICAL CONSOLE LINE:
00000000 0004 200 :
00000000 0004 201 :
00000010 0000 202 RXDBSC_LOG_WRMFLG = 16 :RETURNING VALUE FOR WARMSTART FLAG
00000011 0000 203 RXDBSC_LOG_CLDFLG = 17 :RETURNING VALUE FOR COLDSTART FLAG

```

OPDRV790  
V04-000

- VAX/VMS 11/790 CONSOLE TERMINAL DRIVER N<sup>7</sup> 16-SEP-1984 01:02:49 VAX/VMS Macro V04-00  
5-SEP-1984 04:11:07 [SYSLOA.SRC]OPDRV790.MAR;1 Page 5 (2)

00000012	0000	204 RXDBSC_LOG_UCODE = 18	:RETURNING MICROCODE VERSION
00000020	0000	205 RXDBSC_LOG_CMDCMP = 32	:CONSOLE COMMAND STRING COMPLETE
00000082	0000	206 RXDBSC_LOG_CMDERR = 130	:CONSOLE COMMAND STRING ERROR
00000030	0000	207 RXDBSC_LOG_SNAP = 48	:SNAPFILE STATUS RETURNED
00000040	0000	208 RXDBSC_LOG_REBOOT = 64	:CONSOLE REBOOT SUCCESSFUL

OPI  
VO

0000 210 .SBTTL CONSOLE RECEIVE DISPATCH VECTOR  
0000 211  
00000000 212 .PSECT SYSLOA, LONG  
0000 213  
0000 214  
0000 215 : THE INTERRUPT ROUTINE ADDRESSES FOR EACH OF THE DEVICES THAT SHARE THE  
0000 RXCS/RXDB REGISTER ARE LOADED INTO THIS TABLE BY THE UNIT INITIALIZATION  
0000 ROUTINE FOR EACH DEVICE. CONSINITIAL SETS THE CELLS FOR THE CONSOLE  
0000 TERMINAL, THE ENVIRONMENTAL MONITOR, THE UNUSED VECTORS, AND THE CARRIER  
0000 CHANGE INTERRUPT TYPE.  
0000 220  
00000000 221 CONSINPDISTAB:  
0000 222 .LONG 0 : ID 0 - CONSOLE TERMINAL  
00000000 223 .LONG 0 : ID 1 - REMOTE SERVICE PORT  
00000000 224 .LONG 0 : ID 2 - ENVIRONMENTAL MONITOR  
00000000 225 .LONG 0 : ID 3 - LOGICAL CONSOLE  
00000000 0010 226 .LONG 0[11] : IDs 4-14 - UNUSED, RESERVED TO DIGITAL  
00000000 0020  
00000000 0030  
00000000 003C 227 .LONG 0 : ID 15 - CARRIER CHANGE NOTIFICATION

0040 229 .SBTTL CONSOLE CONTROLLER INITIALIZATION  
 0040 230 ++  
 0040 231 CONSINITIAL - INITIALIZE CONSOLE CONTROLLER  
 0040 232  
 0040 233 FUNCTIONAL DESCRIPTION:  
 0040 234 THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE CONTROLLER.  
 0040 235  
 0040 236 INPUTS:  
 0040 237  
 0040 238 R5 = IDB ADDRESS  
 0040 239 R6 = DDB ADDRESS  
 0040 240 R8 = CRB ADDRESS  
 0040 241  
 0040 242  
 0040 243 OUTPUTS:  
 0040 244  
 0040 245 ALL REGISTERS ARE PRESERVED, EXCEPT R0, R1.  
 0040 246 --  
 10 A8 D5 0040 247 CONSINITIAL: :INITIALIZE CONSOLE INTERFACE  
 50 10 A8 50 0040 248 TSTL CRBSL\_AUXSTRUC(R8) :HAVE WE PASSED THIS WAY BEFORE?  
 0043 249 BNEQ 30\$ :BRANCH IF SO  
 50 50 FFB7 CF 9E 0045 250 MOVAB W^CONSINPDISTAB, R0 :POINT TO THE CONSOLE DISPATCH TABLE  
 10 01AA'CF 50 D0 004A 251 MOVL R0, CRBSL\_AUXSTRUC(R8) :SET POINTER IN CRB  
 60 80 80 D0 004E 252 MOVAB W^INT\_CONSOLINP, (R0) :STORE CONSOLE INTERRUPT ROUTINE  
 80 0281'CF 9E 0053 253 MOVL (R0)+, (R0)+ :SAME ROUTINE FOR REMOTE SERVICES PORT  
 80 030C'CF 9E 0056 254 MOVAB W^INT\_EMMINP, (R0)+ :STORE EMM INTERRUPT RTN  
 80 51 0B D0 0060 255 MOVAB W^INT\_LOGINP, (R0)+ :STORE LOGICAL CONSOLE INTERRUPT RTN  
 80 0402'CF 9E 0063 256 MOVL #11, RT :SET NUMBER OF VECTORS TO INITIALIZE  
 F8 51 51 0068 257 MOVAB W^INT\_IGNOREINP, (R0)+ :STORE IGNORE INTERRUPT ROUTINE  
 60 037B'CF 9E 006B 258 10\$: SOBGTR R1, 10\$ :DO ALL  
 006B 259 MOVAB W^INT\_CARCHANGE, (R0) :STORE CARRIER CHANGE INTERRUPT ADDR  
 0070 260  
 50 50 50 21 DB 0070 261 MFPR #PRS\_RXDB, R0 :READ CURRENT RECEIVER STATUS  
 1D A8 F0 8F 78 0073 262 ASHL #<-RXD V\_CARRIER>, R0, R0 :POSITION CARRIER BYTE TO LOW BYTE  
 0E A5 50 90 0078 263 MOVB R0, CRBSB\_DZ\_CARRIER(R8) :SAVE INITIAL STATUS  
 0C 94 007C 264 CLRBL IDBSB\_TT\_ENABLE(R5) :CLEAR TRANSMIT ENABLE MASK  
 0C 90 007F 265 MOVB #<RXS\_M\_EMM!RXS\_M\_LOGCONS>@<-RXS V\_DTR>,- :SETUP INITIAL LOGICAL DTR  
 50 50 1E A8 0081 266 CRBSB\_DZ\_DTR(R8) :GET LOGICAL DTR  
 50 50 1E A8 9A 0083 267 MOVZBL CRBSB\_DZ\_DTR(R8), R0 :POSITION IN REGISTER  
 50 50 10 78 0087 268 ASHL #RXS\_V\_DTR, R0, R0 :SET INTERRUPT ENABLE  
 20 50 C8 008B 269 BISL2 #RXS\_MIE, R0 :ENABLE THE LINES  
 DA 0092 270 MTPR R0, #PRS\_RXCS  
 0095 271  
 0095 272 30\$: RSB  
 05 0095 273 CON\_RETURN: RSB

	0096	276	.SBTTL CONSOLE UNIT INITIALIZATION		
	0096	277	++ CONSINITLINE - INITIALIZE CONSOLE UNIT		
	0096	278	FUNCTIONAL DESCRIPTION:		
	0096	279	THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE UNITS.		
	0096	280	INPUTS:		
	0096	281	R5 = UCB ADDRESS		
	0096	282	R9 = CRB ADDRESS		
	0096	283	OUTPUTS:		
	0096	284	REGISTERS R4,R5 PRESERVED		
	0096	285	--		
	0096	286	CONSINITLINE::		
51	00 44 A5 15	E2	0096	287	BBSS #TTSV_MODEM,UCBSL_DEVDEPEND(R5),28 ;ENSURE DEVICE IS MODEM
	00 00C4 C5 15	E2	0098	288	2\$: BBSS #TTSV_MODEM,UCBSL_TT_DECHAR(R5),48
	00000000'GF	D0	00A1	289	4\$: MOVL G^TTY\$GL_DPT,R1 ;ADDRESS OF CLASS DPT
	50 1E A1	3C	00A8	290	MOVZUL DPTSW_VECTOR(R1),R0 ;LOCATE CLASS DRIVER VECTOR TABLE
	51 50	C0	00AC	291	ADDL R0,R1 ;RELOCATE BASE ADDRESS
	0114 C5 51	D0	00AF	292	MOVL R1,UCBSL_TT_CLASS(R5) ;SET TERMINAL CLASS DRIVER VECTOR
	010C C5 61	D0	00B4	293	MOVL CLASS_GETNXT(R1),UCBSL_TT_GETNXT(R5)
	0110 C5 04	A1	00B9	294	MOVL CLASS_PUTNXT(R1),UCBSL_TT_PUTNXT(R5)
	50 28 A5	D0	00BF	295	MOVL UCBSL_DDB(R5),R0 ;GET DDB ADDRESS
	OC A0 10 A1	D0	00C3	296	MOVL CLASS_DDT(R1),DDBSL_DDT(R0)
	0088 C5 10 A1	D0	00C8	297	MOVL CLASS_DDT(R1),UCBSL_DDT(R5) ;SET DDT ADDRESS IN UCB
	50 00000000'GF	9E	00CE	298	MOVAB G^OPSDPT,R0 ;GET ADDRESS OF CONSOLE DPT
	51 1E A0	3C	00D5	299	MOVZUL DPTSW_VECTOR(R0),R1 ;OFFSET TO CONSOLE PORT VECTOR DISPATCH TABL
0118	C5 50 51	C1	00D9	300	ADDL R1,R0,UCBSL_TT_PORT(R5) ;SET ADDRESS IN UCB
	00DF			301	
	64 A5 10	A8	00DF	302	BISW #UCBSM_ONLINE,UCBSW_STS(R5) ;SET ONLINE
50	01 54 A5	78	00E3	303	ASHL UCBSW_UNIT(R5),#1,R0 ;BUILD UNIT'S BIT MASK
	0106 C5 50	B0	00E8	304	MOVW R0,UCBSW_TT_UNITBIT(R5) ;SAVE IT IN UCB
	5C A5 B5	00ED		305	TSTW UCBSW_REF(CR5) ;REFCOUNT 0?
	08 12	00F0		306	BNEQ 20\$ ;IF NEQ THEN NO SET UP
	50 0114 C5 00	D0	00F2	307	MOVL UCBSL_TT_CLASS(R5),R0 ;ADDRESS OF CLASS VECTOR TABLE
	08 B0 16	00F7		308	JSB @CLASS_SETUP_UCB(R0) ;INITIALIZE THE UCB FOR CONSOLE TERM.
	00FA			309	
	0124 C5 94	00FA		310	
50	50 21	DB	00FE	311	CLRB UCBSB_TT_DS_RCV(R5) ;CLEAR RECEIVE STATUS
	0106 C5 F0 8F	78	0101	312	MFPR #PRS_RXDB,R0 ;READ CURRENT STATUS
	50 93	0106		313	ASHL #<-RXS_V_DTR>,R0,R0 ;POSITION CARRIER BYTE
	06 13	0108		314	BITB R0,UCBSW_TT_UNITBIT(R5) ;CARRIER UP FOR THIS LINE?
	F0 8F 90	010D		315	BEQL 25\$ ;BRANCH IF NOT
	0124 C5 0110			316	MOVB #<TTSM_DS_CARRIER!TTSM_DS_DSR!TTSM_DS_CTS!TTSM_DS_RING>,-
	51 00 9A 0113			317	UCBSB_TT_DS_RCV(R5) ;YES, SET IT AND BITS ALWAYS SET
	5C A5 B5 0116			318	MOVZBL #MODEMSC_INIT,R1 ;SET TO INIT MODEM PROTOCOL
	03 13 0119			319	TSTW UCBSW_REF(CR5) ;ANY CHANNELS ATTACHED?
	50 51 01 9A 011B			320	BEQL 30\$ ;IF EQL NO
	52 DD 0123			321	MOVZBL #MODEMSC_SHUTDOWN,R1 ;FORCE MODEM HANGUP
	OC B0 16 0125			322	MOVL UCBSL_TT_CLASS(R5),R0 ;ADDRESS CLASS VECTOR TABLE
				323	PUSHL R2 ;SAVE R2 OVER DS TRAN
				324	JSB @CLASS_DS_TRAN(R0) ;INVOKE DATA SET-TRANSITION

08 64 A5 52 8ED0 0128 333 POPL R2  
50 0114 C5 05 E1 012B 334 40\$: BBC #UCBSV\_POWER,UCBSW\_STS(R5),50\$ : RESTORE R2  
20 80 17 D0 0130 335 MOVL UCBSEL\_TT\_CLASS(R5)-R0 ; DID WE DETECT A POWERFAIL?  
0138 336 JMP @CLASS\_POWERFAIL(R0) ; YES, GET CLASS VECTOR TABLE ADDRESS  
0138 337 50\$: ; AND GO TO THE POWERFAIL CODE  
0138 338 CONSSET\_LINE::  
0138 339 CONSSET\_MODEM::  
0138 340 CONSNULL::  
05 0138 341 CONSINIT\_CTY::  
RSB

	0139	344	.SBTTL CONSDISCONNECT DISCONNECT LINE			
	0139	345	++ CONSDISCONNECT - SHUT OFF UNIT			
	0139	346				
	0139	347	FUNCTIONAL DESCRIPTION:			
	0139	348	THIS ROUTINE IS USED WHEN FOR SOME REASON THE UNIT MUST BE DISCONNECTED			
	0139	349				
	0139	350				
	0139	351	INPUTS:			
	0139	352	R5 = UCB ADDRESS			
	0139	353	OUTPUTS:			
	0139	354				
	0139	355				
	0139	356	R3,R4 ARE USED			
	0139	357				
	0139	358				
	0139	359	--			
	0139	360	CONSDISCONNECT::			
	0139	361	PUSHR	#^M<R0,R1,R2>	:SET MODEM SHUTDOWN	
50	S1 07	BB 0139	362	MOVL	#MODEMSC_SHUTDOWN,R1	:ACCESS CLASS VECTOR TABLE
	0114 C5	D0 013B	363	MOVL	UCBSL TT_CLASS(R5),R0	:INVOKE MODEM TRANSITION ROUTINE
	OC 80	D0 013E	364	JSB	@CLASS_D5_TRAN(R0)	
	07	BA 0143	365	POPR	#^M<R0,R1,R2>	:RESTORE REGISTERS
	05	0146	366		RSB	
		367				

0149 369 .SBTTL OUTPUT MODEM CONTROL  
0149 370 ::++  
0149 371 : CONSDS\_SET - SET OUTPUT MODEM SIGNALS  
0149 372  
0149 373 : FUNCTIONAL DESCRIPTION:  
0149 374  
0149 375 : THIS ROUTINE OUTPUTS THE OUTPUT MODEM SIGNALS FOR THE SPECIFIED UNIT  
0149 376  
0149 377 : INPUTS:  
0149 378  
0149 379 : R2 = LOW BYTE - SIGNALS TO ACTIVATE  
0149 380 : HIGH BYTE - SIGNALS TO DEACTIVATE  
0149 381  
0149 382 : RS = UCB ADDRESS  
0149 383  
0149 384 : OUTPUTS:  
0149 385  
0149 386 : R0-R3 ARE USED.

```

0185 403 .SBTTL CONSOLE RECEIVER INTERRUPT DISPATCHER
0185 404 ++
0185 405 CONSINTINP - CONSOLE INTERRUPT ON INPUT READY
0185 406
0185 407 FUNCTIONAL DESCRIPTION:
0185 408
0185 409 THIS ROUTINE IS ENTERED AS A RESULT OF A RECEIVER INTERRUPT ON THE
0185 410 CONSOLE INTERFACE. THE INTERRUPT CAN BE GENERATED BY THE CONSOLE
0185 411 TERMINAL, REMOTE SERVICES PORT, ENVIRONMENTAL MONITOR, OR THE
0185 412 11/790 LOGICAL CONSOLE.
0185 413
0185 414 CONSOLE TERMINAL:
0185 415 REMOTE SERVICES PORT: ALL RECEIVED DATA CHARACTERS ARE CONSIDERED
0185 416 UNSOLICITED AND RESULT IN AN ENTRY INTO THE
0185 417 TERMINAL DRIVER COMMON CHARACTER BUFFERING
0185 418 ROUTINE 'BUCBSL_TT_PUTNXT(R5)'.
0185 419
0185 420 ENVIRONMENTAL MONITOR:
0185 421
0185 422 REMOTE SERVICES PORT: INTERRUPT CAN EITHER BE AN ALERT FOR TEMPERATURE
0185 423 OR BLOWER, OR A PERIODIC SAMPLING.
0185 424
0185 425 LOGICAL CONSOLE:
0185 426
0185 427
0185 428 TEST FOR NOTIFICATION THAT THE CONSOLE WAS REBOOTED
0185 429 SUCCESSFULLY; IF SO, LOG THE EVENT. THERE IS
0185 430 CURRENTLY NO OTHER FULL DRIVER SUPPORT FOR THE LOGICAL
0185 431 CONSOLE LINE.
0185 432
0185 433 INPUTS:
0185 434 R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.
0185 435
0185 436 00(SP) = ADDRESS OF THE IDB
0185 437
0185 438 OUTPUTS:
0185 439
0185 440 THE SAVED REGISTERS ARE RESTORED BEFORE REI.
0185 441
0185 442 -- .ENABLE LOCAL_BLOCK
0185 443
0185 444
0185 445 CONSINTINP:: MOVE DATA FROM INTERFACE
0185 446 MFPR #PRS_RXDB,R3
0188 447
0188 448 GET THE ASSOCIATED UCB
0188 449
0188 450 MOVL a(SP)+,R4 :GET IDB ADDRESS
0188 451 ASHL #8,R3,R2 :GET LINE NUMBER
0188 452 BICW #^C<^XF>,R2 :
0188 453
0195 454 SEE IF A UCB IS ASSOCIATED WITH THE LINE. CURRENTLY, THE LOCAL CONSOLE
0195 455 UCB IS ALWAYS PRESENT, AND SYSGEN SUPPORTS ADDING A UCB FOR THE REMOTE
0195 456 CONSOLE, ENVIRONMENTAL MONITOR (EMM), AND LOGICAL CONSOLE LINES. HOWEVER,
0195 457 THERE IS CURRENTLY NO USER-REQUESTED TRANSFER SUPPORT FOR THE EMM OR
0195 458 LOGICAL CONSOLE IN OPDRV790. BUT WE MUST DISPATCH THE EMM AND LOGICAL
0195 459 CONSOLE INTERRUPTS TO ROUTINES THAT CHECK FOR CERTAIN UNSOLICITED MESSAGES.

```

			0195	460	: SUCH AS ENVIRONMENTAL ALERT CONDITIONS AND "CONSOLE REBOOT SUCCESSFUL"				
			0195	461	: MESSAGES.				
			0195	462					
52	OE	B3	0195	463	BITW	#^XE,R2	: IS IT A DEVICE INTERRUPT?		
			0198	464			: (IDS 0-1 MAY HAVE A UCB ASSOCIATED)		
55	18	A442	07	12	0198	465	: BNEQ 7\$		
			00	019A	466	5\$: MOVL	: IF NEQ NO. SKIP UCB TEST		
			15	13	019F	467	: BEQL 30\$	: GET THE UCB ADDRESS	
50	FESA	CF42	00	B0	17	01A1	468	7\$: MOVAL	: NO UCB - DISMISS INTERRUPT
						01A7	469	JMP 2(R0)	: GET ADDRESS TO VECTOR TO
						01AA	470		: DISPATCH TO PROCESS INTERRUPT
						01AA	471	: CONSOLE TERMINAL INTERRUPT	
						01AA	472		
						01AA	473	INT_CONSOLINP:	
53	53	9A	01AA	474	MOVZBL	R3,R3	: ZERO TOP 3 BYTES		
0110	D5	16	01AD	475	JSB	UCBSL_TT_PUTNXT(R5)	: BUFFER THE CHARACTER		
03	13	01B1	476		BEQL	30\$	: IF EQL THEN NO CHARACTER TO OUTPUT		
0179	30	01B3	477	20\$:	BSBW	CONSSTARTIO	: OUTPUT THE CHARACTER		
0249	31	01B6	478	30\$:	BRW	DISMIS	: GO		
			01B9	479					
			01B9	480	.DISABLE_LOCAL_BLOCK				



```

00000003 0281 525 EMM_V_BYTECOUNT = 3
0281 526
0281 527 : DEFINE FORMAT OF FIRST BYTE OF EMM ALERT MESSAGE:
0281 528
0281 529 <07> - WHEN SET, THIS IS AN EMM ALERT CONDITION, AS OPPOSED TO
0281 530 DATA RETURNED IN RESPONSE TO A REQUEST
0281 531 <06> - WHEN SET, THIS IS AN AUTOMATIC SHUTDOWN CONDITION. IF THE
0281 532 CONDITION IS NOT CLEARED IN A SMALL NUMBER OF MINUTES (1-2),
0281 533 THE CPU WILL BE POWERED DOWN.
0281 534 <05> - RESERVED FOR FUTURE USE. NOT GUARANTEED TO BE ZERO.
0281 535 <04:00> - IDENTIFIES WHICH ALERT CONDITION IS BEING SIGNALLED
0281 536
00000007 0281 537 EMM_V_ALERT = 7
00000006 0281 538 EMM_V_ASD = 6
00000000 0281 539 EMM_V_CONDITION = 0
00000005 0281 540 EMM_S_CONDITION = 5
0281 541
0281 542 : DISPATCH HERE FOR INPUT INTERRUPT FROM EMM
0281 543
0281 544 INT_EMMINP:
6D FB AF 02 E0 0281 545 BBS #EMM_V_IGNOREINP, - ; BRANCH IF WE ARE IGNORING THIS
0286 546 EMM_FLAGS, - ; MESSAGE
16 F6 AF 00 E2 0286 547 BBSS #EMM_IN_PROGRESS, - ; BRANCH IF THIS IS NOT THE FIRST BYTE
0288 548 EMM_FLAGS, - ; OF AN EMM MESSAGE
0288 549 EMM_LAST_BYTE
0288 550
0288 551 : HANDLE THE FIRST BYTE OF AN ENVIRONMENTAL MONITOR ALERT MESSAGE
0288 552
0288 553
0288 554 EMM_FIRST_BYTE:
58 53 07 E1 0288 555 BBC #EMM_V_ALERT,R3,- ; DON'T HANDLE RESPONSES TO EMM REQUESTS
028F 556 EMM_RESPONSE ; YET
05 53 EB AF B4 028F 557 CLRW EMM_BUFFER ; ZERO THE BUFFER
06 E1 0292 0296 558 BBC #EMM_V_ASD,R3,10$ ; BRANCH IF NOT AUTOMATIC SHUTDOWN
0296 559
00 E6 AF 01 E2 0296 560 BBSS #EMM_V_SHUTDOWN,- ; CONDITION
0298 561 EMM_FLAGS,10$ ; FLAG AN AUTOMATIC SHUTDOWN SO IT
DE AF 53 90 0298 562 10$: MOVB R3,EMM_BUFFER ; CAN BE REPORTED TO THE CONSOLE TERM
43 11 029F 563 BRB DISMISS_EMM ; BUFFER THIS BYTE OF DATA
02A1 564
02A1 565 : DISMISS THIS INTERRUPT
02A1 566 : HANDLE LAST BYTE OF AN EMM MESSAGE HERE. WRITE AN ERRORLOG ENTRY AND
02A1 567 : BROADCAST A WARNING TO THE CONSOLE TERMINAL.
02A1 568 EMM_LAST_BYTE:
D9 AF 53 90 02A1 569 MOVB R3,EMM_BUFFER+1 ; PUT THE DATA IN 2ND BYTE OF THE BUFFER
51 12 D0 02A5 570 MOVL #EMBSC_HD_LENGTH+2,R1 ; SIZE OF ERRORLOG BUFFER TO ALLOCATE
00000000 GF 16 02A8 571 JSB G^ERLSALLOCEMB ; ALLOCATE ERRORLOG BUFFER
OF 50 E9 02AE 572 BLBC R0,BROADCAST ; BRANCH IF NO BUFFER AVAILABLE
04 A2 OF B0 02B1 573 MOVW #EMBSC_EMM,EMBSW_HD_ENTRY(R2) ; SET THE ERRORLOG TYPE
02B5 574
10 A2 C5 AF B0 02B5 575 MOVW EMM_BUFFER,EMBSC_HD_LENGTH(R2) ; MOVE THE DATA INTO THE ERRORLOG BUFFER
00000000 GF 16 02BA 576 JSB G^ERLSRELEASEMB ; RELEASE THE ERRORLOG DATA
51 B8 AF 3C 02C0 577 BROADCAST: MOVZWL EMM_MSGLEN,R1 ; LENGTH OF THE MESSAGE
04 B8 AF 01 E5 02C4 579 BBCC #EMM_V_SHUTDOWN, - ; BRANCH IF THIS IS NOT AN AUTOMATIC
02C9 580 EMM_FLAGS,10$ ; SHUTDOWN CONDITION

```

55	51	AD AF	A0	02C9	582	ADDW	SHUTDOWN MSGLEN,R1	: ADD SHUTDOWN MESSAGE TO BROADCAST
	52	FEE8 CF	9E	02CD	583	10\$: MOVAB	EMM MESSAGE, R2	: ADDRESS OF MESSAGE
		00000000 GF	9E	02D2	584	MOVAB	G^OPASUCBO, R5	: SEND IT TO THE CONSOLE TERMINAL
		00000000 GF	16	02D9	585	JSB	G^I0CSBROADCAST	: BROADCAST THE MESSAGE
				02DF	586	EMM_ALERT DONE:		
		00 9D AF	00	E5	587	BBCC	#EMM_IN_PROGRESS,-	: FINISHED WITH THIS EMM MESSAGE
				02E1	588		EMM_FLAGS,DISMIS_EMM	
				02E4	589	DISMIS_EMM:		
		011B	31	02E4	590	BRW	DISMIS	
				02E7	591	:		
				02E7	592	WE GET HERE ONLY IF WE'RE FAIRLY MIXED UP: WE DON'T SUPPORT REQUESTING DATA		
				02E7	593	FROM THE EMM BUT THE EMM LINE IS GIVING US A RESPONSE TO A DATA REQUEST.		
				02E7	594	READ THE BYTCOUNT AND IGNORE THAT MANY BYTES OF DATA FROM THE EMM LINE.		
				02E7	595	:		
	00 95 AF	02	E2	02E7	596	EMM_RESPONSE:		
				02EC	597	BBSS	#EMM_V IGNOREINP, -	: SIGNAL WE'RE GETTING AN EMM RESPONSE
	00 90 AF	03	E2	02EC	598		EMM_FLAGS,10\$	
				02F1	599	10\$: BBSS	#EMM_V BYTECOUNT, -	: SIGNAL THE NEXT BYTE SHOULD BE THE
			F1	11	600		EMM_FLAGS,20\$	: MESSAGE BYTE COUNT
				02F1	601	20\$: BRB	DISMIS_EMM	
				02F3	602			
	06 89 AF	03	E5	02F3	603	EMM_IGNORE DATA:		
				02F8	604	BBCC	#EMM_V BYTECOUNT, -	: BRANCH IF THIS IS NOT THE BYTCOUNT
	83 AF	53	90	02F8	605		EMM_FLAGS,10\$	: BYTE
		E6	11	02FC	606	MOVB	R3,EMM_IGNORECNT	: SAVE NUMBER OF BYTES TO IGNORE
				02FE	607	BRB	DISMIS_EMM	: THAT'S IT FOR THIS BYTE
	FF7D CF	CF	97	02FE	608	10\$:		
		E0	12	0302	609	DECB	EMM_IGNORECNT	: COUNT DOWN ONE MORE BYTE
	00 FF77 CF	02	E5	0304	610	BNEQ	DISMIS_EMM	: BRANCH IF STILL MORE TO IGNORE
				030A	611	BBCC	#EMM_V IGNOREINP, -	: CLEAR THE "IGNORE EMM DATA" FLAG
		D8	11	030A	612		EMM_FLAGS,20\$	
				030A	613	20\$: BRB	DISMIS_EMM	: THAT'S IT

030C	615	.SBTTL LOGICAL CONSOLE INPUT INTERRUPTS	
030C	616	++	
030C	617	INT_LOGINP - HANDLE LOGICAL CONSOLE INPUT INTERRUPTS	
030C	618		
030C	619	FUNCTIONAL DESCRIPTION:	
030C	620		
030C	621	ONLY ONE TYPE OF LOGICAL CONSOLE INTERRUPT-LEVEL INPUT IS CURRENTLY HANDLED:	
030C	622	AN UNSOLICITED MESSAGE SIGNALLING THAT THE CONSOLE WAS RE-BOOTTED SUCCESSFULLY.	
030C	623	THE CONSOLE CAN BE RE-BOOTTED BY THE ROUTINE CONSKEEPALIVE IF IT DETECTS THAT	
030C	624	THE CONSOLE HAS DIED. CONSKEEPALIVE WILL WRITE A MESSAGE TO THE SYSTEM	
030C	625	ERRORLOG NOTING THAT IT HAS ATTEMPTED TO RE-BOOT THE CONSOLE. IF THE	
030C	626	REBOOT IS SUCCESSFUL, THIS ROUTINE WILL WRITE ANOTHER MESSAGE TO THE	
030C	627	ERRORLOG SIGNALLING THE SUCCESSFUL REBOOT.	
030C	628		
030C	629	INPUTS:	
030C	630	R3	- CONTENTS OF PRS_RXDB REGISTER
030C	631	--	
030C	632	INT_LOGINP:	
40 8F 53 91	030C 633	CMPB	R3 #RXDBSC_LOG_REBOOT ; Console reboot message?
1A	0310 634	BNEQ	DISMIS_LOG ; Branch if not.
51 11 00	0312 635	MOVL	#EMBSC HD LENGTH+1,R1 ; Allocate a header plus one byte.
00000000'GF	0315 636	JSB	G^ERLSALLOCUMB ; Allocate space in the errorlog buffer.
OE 50	E9 031B 637	BLBC	R0,DISMIS LOG ; Branch if unable to allocate.
04 A2 11	B0 031E 638	MOVW	#EMBSC CRBT, - ; Set entry type = console reboot.
05 A2 01	0322 639	MOVB	EMBSW HD ENTRY(R2) ; Set flag = reboot successful.
00000000'EF	16 0326 640	JSB	#1,EMBSW HD ENTRY+1(R2) ; Release the errorlog data.
00D3	31 032C 641	ERL\$RELEASEMB	
	032C 642	DISMIS_LOG:	
	643	BRW	DISMIS

				032F	645	.SBTTL START I/O ON CONSOLE INTERFACE
				032F	646	++
				032F	647	CON\$STARTIO - START I/O ON CONSOLE INTERFACE
				032F	648	
				032F	649	FUNCTIONAL DESCRIPTION:
				032F	650	
				032F	651	THIS ROUTINE IS ENTERED TO OUTPUT A CHARACTER TO THE CONSOLE INTERFACE.
				032F	652	THE DATA IS QUEUED AND SUBSEQUENTLY OUTPUT ON THE NEXT READY INTERRUPT.
				032F	653	
				032F	654	A RETURN TO THE CALLER IS DONE TO ENTER A 'WAIT FOR INTERRUPT' STATE.
				032F	655	
				032F	656	
				032F	657	
				032F	658	
				032F	659	R3 = DATA TO OUTPUT
				032F	660	RS = UCB ADDRESS
				032F	661	
				032F	662	
				032F	663	OUTPUTS:
				032F	664	
				032F	665	
				032F	666	
				032F	667	ENABLE LOCAL_BLOCK
				032F	668	CON\$STARTIO::
				032F	669	BGEQ 20\$ :BRANCH IF SINGLE CHARACTER
				032F	670	BISW2 #TTYSM_TANK_BURST,- :FLAG BURST MODE ACTIVE
				032F	671	UCBSW_TT_HOLD(R5)
				032F	672	START_TERM_I0:
				032F	673	10\$: MOVL UCBSL_CRB(R5),R1 :GET CRB ADDRESS
				032F	674	MOVL CRBSL_INTD+VE(CSL_IDB(R1)),R1 :GET IDB ADDRESS
				032F	675	BISB2 UCBSW_TT_UNITBIT(R5),IDBSB_TT_ENABLE(R1) :CREATE NEW XMT ENABLE MSK
				032F	676	MOVZBL IDBSB_TT_ENABLE(R1),R1 :GET ENABLE MASK FOR THIS DEVICE
				032F	677	ASHL #TXS_V_TEM,R1,R1 :POSITION TO FIELD
				032F	678	BISL2 #TXS_MIE!fxs_M_WRTENA,R1 :SET INTERRUPT ENABLE
				032F	679	UCBSM_INT,UCBSQ_STS(R5);FLAG INTERRUPT EXPECTED
				032F	680	
				032F	681	DSBINT #^X14 :MAKE FOLLOWING TEST AND SET ATOMIC
				032F	682	MFPR #PRS_TXCS,RO :READ TRANSMIT STATUS
				032F	683	BBC #TXS_V_RDY,RO,15\$ :WAIT 'TIL WE HAVE A VALID COPY
				032F	684	MTPR R1,#PRS_TXCS :ENABLE THIS DEVICE
				032F	685	
				032F	686	
				032F	687	20\$: MOVB R3,UCBSW_TT_HOLD(R5) :SAVE OUTPUT CHARACTER
				032F	688	BISW2 #TTYSM_TANK_HOLD,- :SIGNAL CHARACTER IN TANK
				032F	689	UCBSW_TT_HOLD(R5)
				032F	690	BRB 10\$ :GO ENABLE INTERRUPT
				032F	691	.DISABLE LOCAL_BLOCK

037B 693 .SBTTL CARRIER CHANGE SENT BY CONSOLE  
 037B 694 ++  
 037B 695 : FUNCTIONAL DESCRIPTION:  
 037B 696 :  
 037B 697 : DISPATCH TO HERE IF THE CONSOLE IS TELLING US ABOUT A CHANGE IN  
 037B 698 : CARRIER STATUS  
 037B 699 :  
 037B 700 :--  
 037B 701 :  
 037B 702 INT\_CARCHANGE:  
 53 55 18 A4 D0 037B 703 MOVL IDBSL\_UCBLST(R4),R5 : GET ADDRESS OF CONSOLE UCB  
 54 24 A5 D0 037F 704 MOVL UCBSL\_CRB(R5),R4 : GET CRB ADDRESS IN R4  
 52 53 F0 BF 78 0383 705 ASHL #16,R3,R3 : POSITION CARRIER BITS FROM RXDB REG  
 52 53 1D A4 8D 0388 706 XORB3 CRBSB\_DZ\_CARRIER(R4),R3,R2 ; FIND CHANGED BITS  
 1D A4 53 90 038D 707 MOVB R3,CRBSB\_DZ\_CARRIER(R4) ; SAVE STATE OF LOGICAL CARRIER  
 0391 708 :  
 0391 709 : PROCESS LINES WITH CARRIER CHANGE  
 0391 710 :  
 51 52 08 00 EA 0391 711 20\$: FFS #0,#8,R2,R1 : FIND A LINE  
 00 52 13 13 0396 712 BEQL 50\$ : IF EQL NO MORE  
 55 52 51 E1 0398 713 BBC R1,R2,30\$ : CLEAR BIT  
 55 18 A5 41 D0 039C 714 30\$: MOVL CRBSL\_INTD+VECL\_IDB(R4),R5 ; GET IDB ADDRESS  
 EA 13 03A0 715 MOVL IDBSL\_UCBLST(R5)[R1],R5 ; GET CORRESPONDING UCB ADDR FOR UNIT  
 05 10 03A5 716 BEQL 20\$ : BRANCH IF NO UCB  
 E6 11 03A7 717 BSBB REPORT\_CARCHANGE : REPORT THE CARRIER CHANGE  
 03A9 718 BRB 20\$ : CONTINUE SCANNING  
 03AB 719 50\$:  
 03AB 720 BR\_DISMIS:  
 0054 31 03AB 721 BRW DISMIS : DISMISS THE INTERRUPT  
 03AE 722 ++  
 03AE 723 : FUNCTIONAL DESCRIPTION:  
 03AE 724 :  
 03AE 725 : THIS ROUTINE REPORTS A CHANGE IN CARRIER STATUS TO THE TERMINAL  
 03AE 726 : CLASS DRIVER  
 03AE 727 :  
 03AE 728 : INPUTS:  
 03AE 729 :  
 03AE 730 R5 = UCB ADDRESS  
 03AE 731 R4 = CRB ADDRESS  
 03AE 732 :  
 03AE 733 : OUTPUTS:  
 03AE 734 :  
 03AE 735 : DATASET TRANSITION REPORTED TO CLASS DRIVER  
 03AE 736 :  
 03AE 737 : REGISTERS R0,R1,R2 DESTROYED, ALL OTHERS PRESERVED  
 03AE 738 :--  
 03AE 739 REPORT\_CARCHANGE:  
 50 2D 38 A5 02 E1 03AE 740 BBC #DEVSV TRM,UCBSL\_DEVCHAR(R5),20\$ : BRANCH IF NOT TERMINAL  
 28 44 A5 15 E1 03B3 741 BBC #TTSV MODEM,UCBSL\_DEVDEPEND(R5),20\$ : BRANCH IF NOT MODEM  
 1D A4 01 51 EF 03B8 742 EXTZV R1,#1-CRBSB\_DZ\_CARRIER(R4),R0 : GET CARRIER BIT FOR THIS LINE  
 01 05 50 F0 03BE 743 INSV R0,#TTSV DS\_CARRIER,#1,-:SET/CLEAR IN UCB  
 0124 C5 03C2 744 UCBSB\_TT\_DS\_RCV(R5)  
 00 8F 89 03C5 745 BISB3 #<TTSM\_DS\_DSRTTSM\_DS\_CTS!TTSM\_DS\_RING>-  
 0124 C5 51 03C8 746 RT UCBSB\_TT\_DS\_RCV(R5)-:SET BITS WHICH ARE ALWAYS SET  
 52 0124 C5 9A 03CC 747 MOVZBL UCBSB\_TT\_DS\_RCV(R5),R2 : GET CURRENT RCV MODEM STATUS  
 51 03 9A 03D1 748 MOVZBL #MODEMSC\_DATASET,R1 : SIGNAL DATASET TRANSITION  
 50 0114 C5 D0 03D4 749 MOVL UCBSL\_TT\_CLASS(R5),R0 : GET CLASS VECTOR TABLE

**OPDRV790  
V04-000**

- VAX/VMS 11/790 CONSOLE TERMINAL DRIVER 16-SEP-1984 01:02:49 VAX/VMS Macro V04-00  
CARRIER CHANGE SENT BY CONSOLE 5-SEP-1984 04:11:07 [SYSLOA.SRC]OPDRV790.MAR;1

Page 20  
(12)

OC	1E	BB	03D9	750	PUSHR	#^M<R1,R2,R3,R4>	SAVE VOLATILE REGISTERS
	BO	16	03DB	751	JSB	@CLASS-DS TRAN(R0)	SIGNAL DATASET TRANSITION
	1E	BA	03DE	752	POPR	#^M<R1,R2,R3,R4>	RESTORE REGISTERS
	05	03E0	753	20\$:	RSB		RETURN TO CALLER

03E1 755 .SBTTL CONSOLE TRANSMITTER INTERRUPT SERVICE  
 03E1 756 :++  
 03E1 757 :CONSINTOUT - CONSOLE TRANSMITTER INTERRUPT SERVICE  
 03E1 758 :FUNCTIONAL DESCRIPTION:  
 03E1 759 :THIS ROUTINE IS ENTERED WHEN A CONSOLE UNIT READY INTERRUPT OCCURS.  
 03E1 760 :THE INTERRUPT STATE OF THE UNIT IS CHECKED FOR EXPECTED INTERRUPT.  
 03E1 761 :IF NO INTERRUPT IS EXPECTED, THE INTERRUPT IS DISMISSED. IF AN INTERRUPT  
 03E1 762 :IS EXPECTED THEN THE DRIVER IS ENTERED. IN THE CASE OF THE CONSOLE TERMINAL,  
 03E1 763 :A SPECIFIC ROUTINE IS ENTERED TO GET THE NEXT CHARACTER AVAILABLE TO OUTPUT  
 03E1 764 :ON THE UNIT.  
 03E1 765 :  
 03E1 766 :  
 03E1 767 :  
 03E1 768 :  
 03E1 769 :  
 03E1 770 :  
 03E1 771 : INPUTS:  
 03E1 772 : R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.  
 03E1 773 : 00(SP) = ADDRESS OF THE IDB  
 03E1 774 :  
 03E1 775 :  
 03E1 776 :  
 03E1 777 :  
 03E1 778 :  
 03E1 779 :  
 50 9E D0 03E1 780 : CONSINTOUT:  
 53 53 22 DB 03E4 780 : MOVL @SP+,R0 :GET ADDRESS OF IDB  
 53 FFFFFFO 8F 78 03E7 781 :  
 55 18 A043 08 13 03F3 782 : Note that R0 contains the address of the IDB throughout this routine.  
 03E4 783 :  
 03E4 784 :  
 03E4 785 :  
 03E4 786 :  
 03E4 787 :  
 03E4 788 :  
 03E4 789 :  
 03E4 790 :  
 03E4 791 :  
 0402 792 : INT IGNOREINP:  
 0402 793 : DISMIS:  
 50 8E 7D 0402 794 :  
 52 8E 7D 0405 795 :  
 54 8E 7D 0408 796 :  
 040B 02 040B 797 :  
 040C 798 :  
 040C 799 : DEVICE IS A TERMINAL, EITHER THE CONSOLE OR THE REMOTE SERVICES PORT  
 040C 800 :  
 040C 801 : INTOUT\_TERM:  
 040C 802 :  
 040C 803 :  
 040C 804 :  
 0109 08 91 040C 805 :  
 54 C5 13 040E 806 :  
 0411 807 :  
 0413 808 :  
 0413 809 :  
 0413 810 :  
 0413 811 :  
 .SBTTL CONSOLE TRANSMITTER INTERRUPT SERVICE  
 CONSINTOUT - CONSOLE TRANSMITTER INTERRUPT SERVICE  
 FUNCTIONAL DESCRIPTION:  
 THIS ROUTINE IS ENTERED WHEN A CONSOLE UNIT READY INTERRUPT OCCURS.  
 THE INTERRUPT STATE OF THE UNIT IS CHECKED FOR EXPECTED INTERRUPT.  
 IF NO INTERRUPT IS EXPECTED, THE INTERRUPT IS DISMISSED. IF AN INTERRUPT  
 IS EXPECTED THEN THE DRIVER IS ENTERED. IN THE CASE OF THE CONSOLE TERMINAL,  
 A SPECIFIC ROUTINE IS ENTERED TO GET THE NEXT CHARACTER AVAILABLE TO OUTPUT  
 ON THE UNIT.  
 INPUTS:  
 R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.  
 00(SP) = ADDRESS OF THE IDB  
 OUTPUTS:  
 THE SAVED REGISTERS ARE RESTORED BEFORE REI.  
 CONSINTOUT:  
 MOVL @SP+,R0 :GET ADDRESS OF IDB  
 Note that R0 contains the address of the IDB throughout this routine.  
 MFPR #PRS TXCS,R3 ;READ STATUS REGISTER  
 ASHL #8,R3,R3 ;EXTRACT ID FIELD  
 BICL2 #^C<^XF>,R3 ;CLEAR UNINTERESTING BITS  
 MOVL IDBSL\_UCBLST(R0)[R3].RS ;GET ADDRESS OF UCB  
 BEQL DISMIS ;BRANCH IF NO UCB  
 CASE R3,- ;DISPATCH TO PROCESS  
 <INTOUT\_TERM,INTOUT\_TERM>,-  
 TYPE=B  
 INT IGNOREINP:  
 DISMIS:  
 MOVQ (SP)+,R0 ;RESTORE REGISTERS  
 MOVQ (SP)+,R2  
 MOVQ (SP)+,R4  
 REI  
 DEVICE IS A TERMINAL, EITHER THE CONSOLE OR THE REMOTE SERVICES PORT  
 INTOUT\_TERM:  
 CHECK FOR BURST ACTIVE ON LINE  
 CMPB #TTYSM\_TANK\_BURST@-8,- ;ONLY BURST ACTIVE?  
 UCBSW TT\_HO[D+1](R5)  
 BEQL CON\_BURST ;YES, CONTINUE BURST  
 LOOK FOR NEXT OUTPUT STATE IN TANK

53 0109 C5 06 00 EA 0413 812 FFS CASE #0,#6\_UCBSU\_TT\_HOLD+1(R5),R3 :DISPATCH  
 041A 813 R3,TYPE=B,<= :SEND PREMPT CHARACTER  
 041A 814 CON\_PREMPT,- :STOP OUTPUT  
 041A 815 CON\_STOP,- :CHAR IN TANK  
 041A 816 CON\_CHAR,- :BURST IN PROGRESS  
 041A 817 CON\_BURST,->  
 041A 818 >  
 0426 819 : NO PENDING DATA - LOOK FOR NEXT CHARACTER  
 0426 820 :  
 64 A5 03 8A 0426 821 : BICB #UCBSM\_TIM!UCBSM\_INT,UCBSW\_STS(R5) ;CLEAR TIMEOUT AND EXPECTED  
 042A 822 :  
 042A 823 : CALL CLASS DRIVER FOR MORE OUTPUT  
 042A 824 :  
 01 FF BF 010C D5 16 042A 825 :  
 010B C5 8F 042E 826 JSB UCBSL\_TT\_GETNXT(R5) :GET THE NEXT CHARACTER  
 0435 827 CASEB UCBSB\_TT\_OUTYPE(R5),#-1,#1 :OPTIMIZE FOR THE SINGLE CHARACTER  
 002B' 0435 828 40\$: .WORD CON\_START\_BURST-40\$ :BY SETTING THE LIMIT TO 1  
 000C' 0437 829 .WORD CON\_RESET\_IE-40\$ :BURST SPECIFIED  
 0439 830 :NONE  
 0439 831 :  
 0439 832 : OUTPUT A CHARACTER TO THE CONSOLE  
 0439 833 :  
 53 53 9A 0439 834 20\$: MOVZBL R3,R3 :ENSURE ALL ZEROES  
 23 53 DA 043C 835 MTPR R3,#PRS\_TXDB :OUTPUT CHARACTER  
 C1 11 043F 836 BRB DISMIS  
 0441 837 :  
 0441 838 : DISABLE OUTPUT ON THIS LINE  
 0441 839 :  
 0441 840 50\$:  
 0441 841 CON\_RESET IE:  
 01 BC 64 A5 01 E0 0441 842 BBS #UCBSV\_INT,- :IF INT EXP. THEN DON'T RESET.  
 OE AO 0106 C5 8A 0443 843 UCBSW\_STS(R5),DISMIS :COULD HAVE BEEN SET DURING CALLBACK  
 51 0E AO 9A 0446 844 BICB2 UCBSW\_TT\_UNITBIT(R5),IDB\$B\_TT\_ENABLE(R0) :CLEAR THIS DEVICE  
 51 51 10 78 0450 845 MOVZBL IDBSB\_TT\_ENABLE(R0),R1 :GET NEW ENABLE MASK BITS  
 51 00008040 8F C8 0454 846 ASHL #TXS\_V\_TEM,R1,R1 :POSITION  
 22 51 DA 045B 847 BISL2 #TXS\_MIE!txs\_m\_wrtena,R1 :SET INTERRUPT ENABLE  
 A2 11 045E 848 MTPR R1,#PRS\_TXCS :TELL CONSOLE  
 0460 849 BRB DISMIS  
 0460 850 :  
 0800 8F A8 0460 851 CON\_START\_BURST:  
 0108 C5 0460 852 BISW #TTYSM\_TANK\_BURST,- :SIGNAL BURST ACTIVE  
 0464 853 UCBSW\_TT\_HOLD(R5)  
 0467 854 :  
 0467 855 : CONTINUE BURST OUTPUT  
 0467 856 :  
 53 011C D5 9A 0467 857 CON\_BURST:  
 23 53 DA 046C 858 MOVZBL UCBSL\_TT\_OUTADR(R5),R3 :OUTPUT NEXT BYTE  
 011C C5 D6 046F 859 MTPR R3,#PRS\_TXDB  
 0120 C5 87 0473 860 INCL UCBSL\_TT\_OUTADR(R5) :UPDATE POINTER  
 07 12 0477 861 DECW UCBSW\_TT\_OUTLEN(R5) :UPDATE COUNT  
 0800 8F AA 0479 862 BNEQ 10\$ :NOT LAST CHARACTER  
 0108 C5 047D 863 BICW #TTYSM\_TANK\_BURST,- :RESET BURST ACTIVE  
 FF7F 31 0480 864 UCBSW\_TT\_HOLD(R5)  
 0483 865 10\$: BRW DISMIS  
 0483 866 :  
 0483 867 : OUTPUT SINGLE CHARACTER  
 0483 868 :

					869 CON_CHAR:	
53	0108 C5	9A	0483	870	MOVZBL UCBSW_TT_HOLD(R5),R3	:OUTPUT CHAR IN TANK
23	53	DA	0488	871	MTPR R3_NPRS_TXDB	
0400 8F	AA	048B	872	BICW #TTYSM_TANK_HOLD -	:SHOW TANK EMPTY	
0108 C5	FF6D	31	0492	873	UCBSW_TT_HOLD(R5)	
				874	BRW DISMIS	
				875	;	
				876	: STOP THE OUTPUT	
				877	;	
				878	CON_STOP:	
64	03	8A	0495	879	BICB #UCBSM_INT!UCBSM_TIM,-	
A5			0497	880	UCBSW_STS(R5)	:RESET OUTPUT ACTIVE
A6	11		0499	881	BRB CON_RESET_IE	:DISABLE INTERRUPT THIS DEVICE
				882	;	
				883	: SEND XON OR XOFF CHARACTER	
				884	;	
				885	CON_PREMPT:	
				886	BICW #TTYSM_TANK_PREMPT,-	
0100 8F	AA	0498	887		UCBSW_TT_HOLD(R5)	:RESET XOFF STATE
0108 C5		049F	888		UCBSB_TT_PREMPT(R5),NPRS_TXDB	:OUTPUT THE CHARACTER
23	010A C5	DA	04A2	889	BRW DISMIS	:AND DISMISS THE INTERRUPT

```

04AA 891 SBTTL CONSOLE PORT ACTION ROUTINES
04AA 892 ++
04AA 893 CONSXOFF - SEND XOFF
04AA 894 CONSXON - SEND XON
04AA 895 CONSSTOP - STOP OUTPUT
04AA 896 CONSABORT - ABORT CURRENT OUTPUT
04AA 897 CONSRESUME - RESUME STOPPED OUTPUT
04AA 898
04AA 899 FUNCTIONAL DESCRIPTION:
04AA 900
04AA 901 THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
04AA 902 CONTROL OUTPUT ON THE PORT
04AA 903
04AA 904 INPUTS:
04AA 905
04AA 906 R5 = UCB ADDRESS
04AA 907
04AA 908 OUTPUTS:
04AA 909
04AA 910 R5 = UCB ADDRESS
04AA 911 --
04AA 912 CONSXOFF::
04AA 913 CONSXON:::
0108 C5 0100 8F A8 04AA 914 BISW #TTYSM_TANK_PREMPT,UCBSW_TT_HOLD(R5) ;SCHEDULE XOFF/XON
010A C5 53 90 04B1 915 MOVB R3,UCBSB_TT_PREMPT(R5) ;SAVE THE CHARACTER IN THE PREMPT SLOT
03 64 A5 01 E0 04B6 916 BBS #UCBSV_INT,UCBSW_STS(R5),10$ ;IF OUTPUT ACTIVE, THEN DONE
04BB 917
FE7A 30 04BB 918 BSBW START_TERM_IO ;ENABLE OUTPUT INTERRUPTS
05 04BE 919 10$: RSB
04BF 920
0200 8F A8 04BF 921 CONSSTOP:::
0108 C5 05 04C3 922 BISW #TTYSM_TANK_STOP - ;SCHEDULE STOP
04C6 923 RSB
04C7 924
04C7 925
0108 C5 08 04C7 926 CONSABORT:::
00 E5 04C7 927 BBCC #TTYSV_TANK_BURST,UCBSW_TT_HOLD(R5),- ;RESET BURST ACTIVE
04CC 928 10$:
04CD 929 10$: TIMSET 1 ;SET A TIMEOUT
04CD 930 RSB ;IN CASE OUTPUT ACTIVE
04E0 931
05 04E0 932 RSB
04E1 933
04E1 934 CONSRESUME:::
0108 C5 0200 8F DD 04E1 935 PUSHL R1 ;SAVE A REGISTER
AA 04E3 936 BICW #TTYSM_TANK_STOP-
04EA 937 UCBSW_TT_HOLD(R5)
21 0108 C5 08 E1 04EA 938 BBC #TTYSV_TANK_BURST,UCBSW_TT_HOLD(R5),20$ ;BRANCH IF NO BURST IN PROGR
51 0120 C5 3C 04F0 939 MOVZWL UCBSW_TT_OUTLEN(R5),R1 ;NUMBER OF BURST CHARS
04F5 940 TIMSET R1 R1 ;SET THE TIMER
13 11 050F 941 BRB 30$ ;CHAR IN TANK OR OTHER TIMEOUT
0511 942
03 64 A5 01 F0 0511 943 20$: TIMSET 2 ;CHAR IN TANK OR OTHER TIMEOUT
FEOC 30 0524 944 30$: BBS #UCBSV_INT,UCBSW_STS(R5),40$ ;SKIP IF OUTPUT ON
0529 945 BSBW START_TERM_IO ;ENABLE OUTPUT INTERRUPT
052C 946 40$: POPL R1
51 BED0 052C 947

```

OPDRV790  
V04-000

- VAX/VMS 11/790 CONSOLE TERMINAL DRIVER 16-SEP-1984 01:02:49 VAX/VMS Macro V04-00  
CONSOLE PORT ACTION ROUTINES 5-SEP-1984 04:11:07 [SYSLOA.SRC]OPDRV790.MAR;1 Page 25  
H 9 (14)

05 052F 948 RSB

0530 950 .SBTTL CONSENDCONSCMD - Send command to 11/790 console

0530 951 ++ Functional Description:  
CONSENDCONSCMD is used to send a command to the "logical" console,  
such as "Examine warm start flag" or "Reboot CPU". It is also  
used by code at IPL 31 to read data in console memory  
(as opposed to using the logical console QIO interface).

0530 952 Inputs:  
R0 = code for console command (see TXDBSC\_LOG\_xxx definitions)  
R2 = # of bytes of data expected (if 0 then just send command)  
R3 = address of buffer to store data in (only if R2 is non-zero)

0530 953 Outputs:  
Data is stored in the buffer.  
All registers preserved.

0530 954 :--

0530 955 CONSENDCONSCMD::  
OF BB 0530 956 PUSHR #^M<R0,R1,R2,R3> ; Save working registers.

0530 957 0532 958 : Enable transmit/receive on logical console line.

04 AE 7E 7C 0532 959 0532 960 0532 961 0532 962 0532 963 0532 964 0532 965 0532 966 0532 967 0530 968 0530 969 0530 970 0530 971 0530 972 0530 973 0530 974 0530 975 0530 976 0530 977 0530 978 0530 979 0530 980 0530 981 0530 982 0530 983 0530 984 0530 985 0530 986 0530 987 0530 988 0530 989 0530 990 0530 991 0530 992 0530 993 0530 994 0530 995 0530 996 0530 997 0530 998 0530 999 0530 1000 0530 1001 0530 1002 0530 1003 0530 1004 0530 1005 0530 1006 CLRQ -(SP) ; Save RXCS and TXCS on stack.  
MFPR #PRS\_RXCS,4(SP) ; Save receive status register.  
MFPR #PRS\_TXCS,(SP) ; Save transmit status register.  
BBC #TXS\_V\_RDY,(SP),20\$ ; Make sure we have a valid copy.  
BICL #^C<TXS\_M\_TEM!TXS\_M\_IE>,- ; Only save writable fields of TXCS  
(SP) ; (transmit enable mask and int enable).  
BISL #TXS\_M\_WRTENA,(SP) ; Set "write enable" in saved TXCS.  
MTPR #TXS\_M\_LOGCONS!TXS\_M\_WRTENA, - ; Disable all console lines except  
#PRS\_RXCS ; logical console; also disable interrupts.  
MTPR #0,#PRS\_RXCS ; Disable interrupts in RXCS.

0530 984 : Send command to logical console.

0530 985 40\$: MFPR #PRS\_RXCS,R1 ; Get transmit status.  
BBC #TXS\_V\_RDY,R1,40\$ ; Loop until ready bit is set.  
CMPZV #TXS\_V\_ID,#TXS\_S\_ID,R1,#3 ; ID = logical console data?  
BNEQ 30\$ ; If not, try again.  
02 50 91 0565 990 CMPB R0,#CONSC\_BOOTCPU ; Console function = boot CPU?  
23 50 13 0568 991 BEQL 90\$ ; Branch if so.  
056A 992 MTPR R0,#PRS\_RXDB ; Else send request code to console.

056D 993 : Get returned data (if any is expected).

056D 994 50\$: TSTL R2 ; Any return data expected?  
BEQL 60\$ ; None; we're all through.  
F9 51 20 52 D5 056D 996 056D 997 056D 998 056D 999 056D 1000 056D 1001 056D 1002 056D 1003 056D 1004 056D 1005 056D 1006 BEQL 60\$ ; Get receiver status.  
MFPR #PRS\_RXCS,R1 ; Loop until done bit is set.  
BBC #RXS\_V\_DONE,R1,50\$ ; Get received data.  
MFPR #PRS\_RXDB,R1 ; ID = logical console data?  
CMPZV #RXD\_V\_ID,#RXD\_S\_ID,R1,#3 ; If not, throw data away and try again.  
BNEQ 50\$ ; Code = requested data?  
83 51 04 08 057B 1001 CMPB R1,R0 ; No recovery from protocol error.  
EF 12 0580 1002 BNEQ CONSOLE\_ERROR ; Put data byte in user's buffer.  
0582 1003 MOVBL R1,(R3)+ ; Branch back to get another byte.

			0588	1007			
			0588	1008	:		
			0588	1009	: Restore console IPRs and registers, and return.		
			0588	1010			
F9	51	22	DB	0588	1011	60\$: MFPR #PRS-TXCS,R1	: Read transmit status.
	51	07	E1	0588	1012	BBC #TXS-V RDY,R1,60\$	: Wait for ready.
22	8E	DA	058F	1013	MTPR (SP)‡,#PRS-TXCS	: Restore previous state of TXCS.	
20	8E	DA	0592	1014	MTPR (SP)+ #PRS-RXCS	: Restore previous state of RXCS.	
	0F	BA	0595	1015	POPR #^M<R0,R1,R2,R3>	: Restore registers.	
			05	0597	1016	RSB	
					0598 1017		
					0598 1018	:	
					0598 1019	: Come here to reboot CPU.	
					0598 1020	:	
23	50	DA	0598	1021	90\$: MTPR R0,#PRS-TXDB	: Send reboot command.	
			00	059B	1022	HALT	: Halt to let reboot occur.

	059C	1024		.SBTTL "ALLOCATE" CONSOLE TERMINAL
	059C	1025		
	059C	1026	++	
	059C	1027		CONSOWNCTY - "ALLOCATE" CONSOLE TERMINAL
	059C	1028		
	059C	1029		FUNCTIONAL DESCRIPTION:
	059C	1030		
	059C	1031		THIS ROUTINE SHOULD BE CALLED WHEN PERFORMING NON-INTERRUPT DRIVEN
	059C	1032		I/O TO THE CONSOLE TERMINAL. IT DISABLES INTERRUPTS AND DOES ANY
	059C	1033		CPU-SPECIFIC INITIALIZATION OF THE CONSOLE TERMINAL REGISTERS.
	059C	1034		THE CONSOLE TERMINAL IS RESTORED TO ITS PREVIOUS STATE BY CALLING
	059C	1035		CONSRELEASECTY.
	059C	1036		
	059C	1037		THIS ROUTINE SHOULD BE CALLED AT OR ABOVE IPL 20.
	059C	1038		
	059C	1039		*** NOTE *** THE CALLER IS RESPONSIBLE FOR RESTORING THE STATE
	059C	1040		OF THE CONSOLE TERMINAL STATUS REGISTERS (RXCS AND
	059C	1041		TXCS) BY CALLING CONSRELEASECTY WITH THE VALUES
	059C	1042		RETURNED BY THIS ROUTINE.
	059C	1043		
	059C	1044		INPUTS:
	059C	1045		NONE
	059C	1046		
	059C	1047		OUTPUTS:
	059C	1048		
	059C	1049		R0: VALUE TO BE RESTORED TO TXCS WHEN CALLING CONSRELEASECTY
	059C	1050		R1: VALUE TO BE RESTORED TO RXCS WHEN CALLING CONSRELEASECTY
	059C	1051		
	059C	1052		PRS_RXCS AND PRS_TXCS ARE SET UP SO THAT NON-INTERRUPT I/O CAN BE
	059C	1053		PERFORMED TO THE CONSOLE TERMINAL.
	059C	1054		
	059C	1055		11/780, 11/750, AND 11/730:
	059C	1056		CONSOLE INTERRUPTS ARE DISABLED
	059C	1057		
	059C	1058		11/790:
	059C	1059		CONSOLE TRANSMIT AND RECEIVE MASKS ARE SET UP SO THAT ONLY
	059C	1060		I/O TO THE CONSOLE TERMINAL IS PERMITTED. INTERRUPTS ARE
	059C	1061		DISABLED.
	059C	1062	--	
	059C	1063		CONSOWNCTY::
	F9 50 22	DB	059C	1064 10S: MFPR #PRS_TXCS, R0 : GET VALUE TO BE RESTORED TO TXCS.
	F9 50 07	E1	059F	1065 BBC #TXS_V RDY R0, 10S : WAIT FOR VALID COPY
	FF00FFBF 8F	CA	05A3	1066 BICL #^C<TXS_M_TEM!TXS_M_IE>, - : ONLY SAVE WRITABLE FIELDS OF TXCS
	50	05A9	1067	RO : (TRANSMIT ENABLE MASK AND INT ENABLE).
50	00008000 8F	C8	05AA	1068 BISL #TXS_M_WRTENA, R0 : SET WRITE ENABLE BIT FOR MASK.
	00018000 8F	DA	05B1	1069 MTPR #TXS_M_CONSTERM!TXS_M_WRTENA, - : ENABLE LOCAL TERM OUTPUT AND
	22	05B7	1070	#PRS_TXCS : DISABLE TRANSMIT INTERRUPTS.
20	00010000 8F	DA	05B8	1071 MTPR #RXS_M_CONSTERM, #PRS_RXCS : ENABLE LOCAL TERM INPUT AND DISABLE
			05BF	1072 RSB : RECEIVE INTERRUPTS.
			05	05BF 1073

			05C0 1075	.SBTTL RELEASE CONSOLE TERMINAL		
			05C0 1076	++		
			05C0 1077	CONSRELEASECTY - RELEASE CONSOLE TERMINAL		
			05C0 1078			
			05C0 1079			
			05C0 1080			
			05C0 1081			
			05C0 1082			
			05C0 1083			
			05C0 1084			
			05C0 1085			
			05C0 1086			
			05C0 1087	INPUTS:		
			05C0 1088	R0:	VALUE RETURNED BY CONSOWNCTY TO BE RESTORED TO TXCS	
			05C0 1089	R1:	VALUE RETURNED BY CONSOWNCTY TO BE RESTORED TO RXCS	
			05C0 1090			
			05C0 1091	OUTPUTS:		
			05C0 1092	NONE.		
			05C0 1093			
			05C0 1094	--		
			05C0 1095	CONSRELEASECTY::		
F9	52	S2	DD	PUSHL R2	: SAVE A REGISTER.	
	22	DB	05C2	10\$: MFPR #PRS-TXCS,R2	: READ TRANSMIT STATUS.	
	07	E1	05C5	BBC #TXS-V RDY,R2,10\$	: WAIT TIL INTERFACE IS READY.	
	50	DA	05C9	MTPR R0,#PRS-TXCS	: RESTORE TXCS TO PREVIOUS STATE.	
	51	DA	05CC	MTPR R1,#PRS-RXCS	: RESTORE RXCS TO PREVIOUS STATE.	
	52	8E	DO	05CF	MOVL (SP)+,R2	: RESTORE THE REGISTER.
			05	05D2	RSB	
			1101			
			1102			

05D3 1104 .SBTTL CON\$GETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL  
05D 1105 ++  
05D 1106 CON\$GETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL  
05D 1107  
05D 1108 : FUNCTIONAL DESCRIPTION:  
05D 1109  
05D 1110 THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O  
05D 1111 DIRECTLY TO THE CONSOLE TERMINAL  
05D 1112  
05D 1113 : INPUTS:  
05D 1114 None  
05D 1115  
05D 1116 : OUTPUTS:  
05D 1117 R0 contains the character.  
05D 1118  
05D 1119 :--  
00000013 05D 1120 control\_s = 19 : control s (xon)  
00000011 05D 1121 control\_q = 17 : control q (xon)  
05D 1122  
05D3 1123 CON\$GETCHAR::  
F9 50 20 DB 05D3 1124 10\$: mfpr #pr\$\_rxcs,r0 ; receiver ready?  
50 07 E1 05D6 1125 bbc #rxs\_v\_done,r0,10\$ ; if clr, receiver not ready  
50 21 DB 05DA 1126 mfpr #pr\$\_rxdb,r0 ; read input character  
05 05 05DD 1127 rsb ; return

05DE 1129 .SBTTL CONSPUTCHAR - PUT A CHARACTER TO THE CONSOLE TERMINAL  
 05DE 1130 ++ CONSPUTCHAR - PUT A CHARACTER TO THE CONSOLE TERMINAL  
 05DE 1131  
 05DE 1132  
 05DE 1133  
 05DE 1134  
 05DE 1135  
 05DE 1136  
 05DE 1137  
 05DE 1138  
 05DE 1139  
 05DE 1140  
 05DE 1141  
 05DE 1142  
 05DE 1143  
 05DE 1144  
 05DE 1145 -- CONSPUTCHAR::  
 1B 51 51 20 DD 05DE 1146 pushl r1 ;save a register  
 13 51 07 07 E1 05E0 1147 mfpr #pr\$\_rxcs,r1 ;receiver ready?  
 51 21 DB 05E3 1148 bbc #rxs\_v\_done,r1,30\$ ;if clr, receiver not ready  
 F9 51 07 00 ED 05EA 1149 mfpr #pr\$\_rxdb,r1 ;read input character.  
 51 11 12 05EF 1150 cmpzv #0,#7,r1,#control\_s ;control-s?  
 11 51 07 20 DB 05F1 1151 bneq 30\$ ;if neq no  
 F9 51 07 E1 05F4 1152 20\$: mfpr #pr\$\_rxcs,r1 ;receiver ready?  
 51 21 DB 05F8 1153 bbc #rxs\_v\_done,r1,20\$ ;if clr, receiver not ready  
 51 00 ED 05FB 1154 mfpr #pr\$\_rxdb,r1 ;read input character  
 EF 12 0600 1155 cmpzv #0,#7,r1,#control\_q ;is it a control-q?  
 F9 51 22 DB 0602 1156 bneq 20\$ ;no, wait for another character.  
 23 51 07 E1 0605 1157 30\$: mfpr #pr\$\_txcs,r1 ;transmitter done?  
 50 DA 0609 1158 bbc #txs\_v\_rdy,r1,30\$ ;if clr, transmitter not done  
 51 8ED0 060C 1160 mtpr r0,#pr\$\_txdb ;write output character  
 05 060F 1161 popl r1 ;restore a register  
 0610 1162 rsb ;return  
 0610 1163 .END

BROADCAST	000002C0	R	02	EMM_FLAGS	00000280	R	02
BR_DISMIS	000003AB	R	02	EMM_IGNORECNT	0000027F	R	02
CLASS_DDT	= 00000010			EMM_IGNORE_DATA	= 000002F3	R	02
CLASS_DS_TRAN	= 0000000C			EMM_IN_PROGRESS	= 00000000		
CLASS_GETNXT	= 00000000			EMM_LAST_BYTE	= 000002A1	R	02
CLASS_POWERFAIL	= 00000020			EMM_MESSAGE	000001B9	R	02
CLASS_PUTNXT	= 00000004			EMM_MSGLEN	0000027B	R	02
CLASS_SETUP_UCB	= 00000008			EMM_RESPONSE	= 000002E7	R	02
CONSABORT	= 000004C7	RG	02	EMM_S_CONDITION	= 00000005		
CON\$BOOTCPU	= 00000002			EMM_V_ALERT	= 00000007		
CON\$DISCONNECT	00000139	RG	02	EMM_V_ASD	= 00000006		
CON\$DS_SET	00000149	RG	02	EMM_V_BYTECOUNT	= 00000003		
CON\$GETCHAR	000005D3	RG	02	EMM_V_CONDITION	= 00000000		
CON\$INITIAL	00000040	RG	02	EMM_V_IGNOREINP	= 00000002		
CON\$INITLINE	00000096	RG	02	EMM_V_SHUTDOWN	= 00000001		
CON\$INIT_CTY	00000138	RG	02	ERL\$ALLOCUMB	*****	X	02
CON\$INPDISTAB	00000000	R	02	ERL\$RELEASEMB	*****	X	02
CON\$INTINP	00000185	RG	02	EXE\$GL_ABSTIM	*****	X	02
CON\$INTOUT	000003E1	RG	02	IDBSB_TT_ENABLE	= 0000000E		
CON\$NULL	00000138	RG	02	IDBSL_UCBLST	= 00000018		
CON\$OWNCTY	0000059C	RG	02	INTOUT_TERM	0000040C	R	02
CON\$PUTCHAR	000005DE	RG	02	INT_CARCHANGE	0000037B	R	02
CON\$RELEASECTY	000005C0	RG	02	INT_CONSOLINP	000001AA	R	02
CON\$RESUME	000004E1	RG	02	INT_EMMINP	00000281	R	02
CON\$SENDCONSCMD	00000530	RG	02	INT_IGNOREINP	00000402	R	02
CON\$SET_LINE	00000138	RG	02	INT_LOGINP	0000030C	R	02
CON\$SET_MODEM	00000138	RG	02	IOC\$BROADCAST	*****	X	02
CON\$STARTIO	0000032F	RG	02	MODEMSC_DATASET	= 00000003		
CON\$STOP	000004BF	RG	02	MODEMSC_INIT	= 00000000		
CON\$XOFF	000004AA	RG	02	MODEMSC_SHUTDOWN	= 00000001		
CON\$XON	000004AA	RG	02	OP\$DPT	*****	X	02
CONTROL_Q	= 00000011			OP\$AUCBO	*****	X	02
CONTROL_S	= 00000013			PRS_IPL	= 00000012		
CON_BURST	00000467	R	02	PRS_RXCS	= 00000020		
CON_CHAR	00000483	R	02	PRS_RXDB	= 00000021		
CON_PREMPT	0000049B	R	02	PRS_TXCS	= 00000022		
CON_RESET IE	00000441	R	02	PRS_TXDB	= 00000023		
CON_RETURN	00000095	R	02	REPORT_CARCHANGE	000003AE	R	02
CON_START_BURST	00000460	R	02	RXD	00000000		
CON_STOP	00000495	R	02	RXDBSC_LOG_CLDFLG	= 00000011		
CRBSB_DZ_CARRIER	= 0000001D			RXDBSC_LOG_CMDCMP	= 00000020		
CRBSB_DZ_DTR	= 0000001E			RXDBSC_LOG_CMDERR	= 00000082		
CRBSL_AUXSTRUC	= 00000010			RXDBSC_LOG_REBOOT	= 00000040		
CRBSL_INTD	= 00000024			RXDBSC_LOG_SNAP	= 00000030		
DBBSL_DDT	= 0000000C			RXDBSC_LOG_UCODE	= 00000012		
DEVSV_TRM	= 00000002			RXDBSC_LOG_WRMFLG	= 00000010		
DISMIS	00000402	R	02	RXD_S_ID	= 00000004		
DISMIS_EMM	000002E4	R	02	RXD_V_CARRIER	= 00000010		
DISMIS_LOG	0000032C	R	02	RXD_V_ID	= 00000008		
DPT\$W_VECTOR	= 0000001E			RXS	00000000		
EMBSC_CGBT	= 00000011			RXS_M_CONSTTERM	= 00010000		
EMBSC_EMM	= 0000000F			RXS_M_EMM	= 00040000		
EMBSC_HD_LENGTH	= 00000010			RXS_M_IE	= 00000040		
EMBSW_HD_ENTRY	= 00000004			RXS_M_LOGCONS	= 00080000		
EMM_ALERT_DONE	000002DF	R	02	RXS_V_DONE	= 00000007		
EMM_BUFFER	0000027D	R	02	RXS_V_DTR	= 00000010		
EMM_FIRST_BYTE	0000028B	R	02	SHUTDOWN_MESSAGE	00000235	R	02

SHUTDOWN_MSGLEN	= 00000279	R 02	UCBSM_TIM	= 00000001
SIZ..	= 00000008		UCBSV_INT	= 00000001
START_TERM_IO	= 00000338	R 02	UCBSV_POWER	= 00000005
TTSM_DS_CARRIER	= 00000020		UCBSW_REF_C	= 0000005C
TTSM_DS_CTS	= 00000010		UCBSW_STS	= 00000064
TTSM_DS_DSR	= 00000080		UCBSW_TT_HOLD	= 00000108
TTSM_DS_RING	= 00000040		UCBSW_TT_OUTLEN	= 00000120
TTSV_DS_CARRIER	= 00000005		UCBSW_TT_PRTCTL	= 00000122
TTSV_DS_DTR	= 00000001		UCBSW_TT_UNITBIT	= 00000106
TTSV_MODEM	= 00000015		UCBSW_UNIT	= 00000054
TTYSGL_DPT	*****	X 02	VECSL_IDB	= 00000008
TTYSM_TANK_BURST	= 00000800			
TTYSM_TANK_HOLD	= 00000400			
TTYSM_TANK_PREMPT	= 00000100			
TTYSM_TANK_STOP	= 00000200			
TTYSV_PC_NOTIME	= 00000000			
TTYSV_TANK_BURST	= 0000000B			
TXDBSC_EMM_CANCEL	= 00000011			
TXDBSC_EMM_ENV	= 00000001			
TXDBSC_EMM_MARGIN	= 00000010			
TXDBSC_EMM_STATUS	= 00000000			
TXDBSC_LOG_CANCEL	= 00000070			
TXDBSC_LOG_EXCOLD	= 00000011			
TXDBSC_LOG_EXUCODE	= 00000012			
TXDBSC_LOG_EXWARM	= 00000010			
TXDBSC_LOG_INVSNP1	= 00000031			
TXDBSC_LOG_INVSNP2	= 00000032			
TXDBSC_LOG_REQERL	= 00000030			
TXDBSC_LOG_SNDDIAG	= 00000020			
TXS	= 00000000			
TXS_M_CONTERM	= 00010000			
TXS_M_IE	= 00000040			
TXS_M_LOGCONS	= 00080000			
TXS_M_TEM	= 00FF0000			
TXS_M_WRTENA	= 00008000			
TXS_S_ID	= 00000004			
TXS_V_ID	= 00000008			
TXS_V_RDY	= 00000007			
TXS_V_TEM	= 00000010			
UCBSB_TT_DS_RCV	= 00000124			
UCBSB_TT_DS_TX	= 00000125			
UCBSB_TT_OUTTYPE	= 0000010B			
UCBSB_TT_PREMPT	= 0000010A			
UCBSL_CRB	= 00000024			
UCBSL_DDB	= 00000028			
UCBSL_DDT	= 00000088			
UCBSL_DEVCHAR	= 00000038			
UCBSL_DEVDEPEND	= 00000044			
UCBSL_DUETIM	= 0000006C			
UCBSL_TT_CLASS	= 00000114			
UCBSL_TT_DECHAR	= 000000C4			
UCBSL_TT_GETNXT	= 0000010C			
UCBSL_TT_OUTADR	= 0000011C			
UCBSL_TT_PORT	= 00000118			
UCBSL_TT_PUTNXT	= 00000110			
UCBSM_INT	= 00000002			
UCBSM_ONLINE	= 00000010			

```
+-----+
! Psect synopsis !
+-----+
```

## PSECT name

	Allocation	PSECT No.	Attributes																	
ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE							
\$ABSS	00000004 ( 4.)	01 ( 1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE							
SYSLOA	00000610 ( 1552.)	02 ( 2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	LONG							

```
+-----+
! Performance indicators !
+-----+
```

## Phase

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.08	00:00:03.38
Command processing	114	00:00:00.45	00:00:03.01
Pass 1	542	00:00:14.97	00:01:10.61
Symbol table sort	0	00:00:02.30	00:00:08.14
Pass 2	211	00:00:03.19	00:00:12.01
Symbol table output	22	00:00:00.14	00:00:00.94
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	923	00:00:21.16	00:01:38.11

The working set limit was 1800 pages.

123886 bytes (242 pages) of virtual memory were used to buffer the intermediate code.

There were 120 pages of symbol table space allocated to hold 2143 non-local and 57 local symbols.

1163 source lines were read in Pass 1, producing 17 object records in Pass 2.

64 pages of virtual memory were used to define 61 macros.

```
+-----+
! Macro library statistics !
+-----+
```

## Macro library name

Macro library name	Macros defined
\$255\$DUA2B:[SYSLOA.OBJ]790DEF.MLB;1	0
\$255\$DUA2B:[SYS.OBJ]LIB.MLB;1	28
\$255\$DUA2B:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	37

2479 GETS were required to define 37 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:OPDRV790/OBJ=OBJ\$:OPDRV790 MSRC\$:OPDRV790/UPDATE=(ENH\$:OPDRV790)+EXECMLS/LIB+LIB\$:790DEF/LIB

0398 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

